

Article

Real-Time Embedded Simulation Platform for Hippocampal Traveling Waves of Electric Field Conduction

Xile Wei ¹ , Zeyu Ren ¹, Meili Lu ² and Siyuan Chang ^{1,*} 

¹ The Tianjin Key Laboratory of Process Measurement and Control, School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China; xilewei@tju.edu.cn (X.W.); 2021234188@tju.edu.cn (Z.R.)

² School of Information Technology Engineering, Tianjin University of Technology and Education, Tianjin 300074, China; meililu@tute.edu.cn

* Correspondence: changsiyuan@tju.edu.cn

Abstract: The investigation of hippocampal traveling waves has gained significant importance in comprehending and treating neural disorders such as epilepsy, as well as unraveling the neural mechanisms underlying memory and cognition. Recently, it has been discovered through both in vivo and in vitro experiments that hippocampal traveling waves are typically characterized by the coexistence of fast and slow waves. However, electrophysiological experiments face limitations in terms of cost, reproducibility, and ethical considerations, which hinder the exploration of the mechanisms behind these traveling waves. Model-based real-time virtual simulations can serve as a reliable alternative to pre-experiments on hippocampal preparations. In this paper, we propose a real-time simulation method for traveling waves of electric field conduction on a 2D plane by implementing a hippocampal network model on a multi-core parallel embedded computing platform (MPEP). A numerical model, reproducing both NMDA-dependent fast waves and Ca-dependent slow waves, is optimized for deployment on this platform. A multi-core parallel scheduling policy is employed to address the conflict between model complexity and limited physical resources. With the support of a graphical user interface (GUI), users can rapidly construct large-scale models and monitor the progress of real simulations. Experimental results using MPEP with four computing boards and one routing board demonstrate that a hippocampal network with a 200 × 16 pyramidal neuron array can execute real-time generation of both fast and slow traveling waves with total power consumption below 500 mW. This study presents a real-time virtual simulation strategy as an efficient alternative to electrophysiological experiments for future research on hippocampal traveling waves.

Keywords: traveling waves; hippocampal network; electric field; real-time simulation; embedded system



Citation: Wei, X.; Ren, Z.; Lu, M.; Chang, S. Real-Time Embedded Simulation Platform for Hippocampal Traveling Waves of Electric Field Conduction. *Electronics* **2024**, *13*, 1130. <https://doi.org/10.3390/electronics13061130>

Academic Editor: Alexander Barkalov

Received: 19 January 2024

Revised: 7 March 2024

Accepted: 15 March 2024

Published: 20 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Hippocampal traveling waves propagate through the hippocampus, a region of the brain associated with learning, memory, and spatial navigation. The generation and propagation of hippocampal traveling waves are believed to play essential roles in information processing and memory consolidation. Experimental studies using techniques such as electrophysiology experiments and functional imaging have provided valuable insights into the mechanisms underlying hippocampal traveling waves. These waves are characterized by synchronized oscillations of neuronal firing and can occur in various frequency and speed ranges, such as theta (4–12 Hz) or gamma (30–100 Hz) frequencies [1]. Through in vitro electrical stimulation experiments, two distinct types of hippocampal traveling waves have been identified. The two types of traveling waves exhibit a significant difference in speed, and they are commonly differentiated based on the terms “fast spikes” and “slow waves” [2]. Further in vitro experiments have revealed that the propagation of these two types of waves does not depend on synaptic transmission but rather resembles electrical field conduction [3,4]. However, these experiments are constrained by factors such as cost,

reproducibility, and ethical limitations. Furthermore, in vitro experiments are prone to accidental errors in experimental conditions and procedural variability, leading to poor reproducibility of results. To address this issue, the academic community has proposed the idea of using model simulations as a substitute for in vitro hippocampal electrical stimulation experiments.

To overcome the limitations of in vitro experiments [1], researchers have conducted modeling and simulation studies on hippocampal arrays to investigate the mechanisms underlying hippocampal traveling waves [5]. By establishing dual-cell models, replication of both fast and slow waves in the hippocampus has been achieved. In larger-scale simulations, the Hodgkin-Huxley neuron model (also abbreviated as H-H model) has been used to replace hippocampal pyramidal neurons, revealing the involvement of calcium ion channels in the generation of hippocampal slow waves [6]. Through the use of membrane potential heatmaps, the propagation process of hippocampal fast spike sequences has been replicated, and the wave sources of fast spikes are believed to be in a continuous motion [3]. These findings have significantly advanced our understanding of hippocampal traveling waves. Further experiments using hippocampal network simulations have been conducted to study the working mechanism of the human brain. A systematic study was conducted on the hippocampal sharp wave by combining model simulation and cognitive map experiments [7]. A small-scale two-dimensional model was established to simulate the state-dependent conduction of hippocampal traveling waves [8]. In silico simulations often require high computational power and memory resources and may not perform well in certain applications that require high real-time performance and convenience. In order to conduct hippocampal electrical stimulation simulations more efficiently, this study attempts to integrate simulation with an embedded semi-physical simulation platform.

A semi-physical simulation platform is a technology used for conducting simulation experiments that combines the characteristics of physical and computational models [9]. It can reduce the cost and risks associated with experiments by avoiding the need for expensive physical equipment and potential damage [10]. Secondly, it provides a more flexible experimental environment that allows for the simulation of various scenarios and conditions by adjusting model parameters and settings [11,12]. It finds wide application in research and development in fields such as transportation, aerospace, and energy [11,12]. In the context of hippocampal research, the semi-physical simulation platform also holds promise as an excellent alternative to in vitro electrical stimulation experiments. Researchers used an in-the-loop simulation system to control the neuron system of a person with tetraplegia [13]. The in-the-loop simulation platform has been deployed to stimulate the temporal lobe of the hippocampus in an attempt to treat brain diseases and improve brain performance [14]. However, there are several challenges associated with applying semi-physical simulation to hippocampal electrical stimulation experiments. The huge scale of hippocampal computing models poses higher demands on the computational power and real-time performance of simulation platforms [15,16].

Many previous studies have attempted to optimize the integration of semi-physical simulation platforms with neuronal models to achieve a better fit. In previous research [17], the authors summarized and reviewed a semi-physical simulation of a flight system on embedded microprocessors. However, specific recommendations were not provided for platform construction for specific microprocessors. In addition to improving the computational capability of individual microprocessors, improving system architecture and enhancing communication and data transmission efficiency can also significantly enhance the computational capability of simulation platforms. Optimizations of the hippocampal array two-compartment model have also been conducted, such as upgrading the model from one-dimensional to two-dimensional and transforming the single-compartment model into a dual-compartment model, making the simulation of the neuronal array closely resemble the physiological functioning of the hippocampus [18]. In a study by [19], the authors discussed several techniques for reducing the computational complexity of H-H model neurons, including network pruning and structural simplification, making these models

more suitable for deployment on resource-constrained devices such as smartphones or embedded systems. Efficient real-time simulation has been achieved under limited computing resources to simulate the behavior of neural dynamics in real-time in embedded systems like FPGA and Raspberry Pi [20,21]. These studies provide insights into the simplification of the hippocampal neuronal array and making it more suitable for small-scale embedded computing platforms.

In this work, we aim to build a hardware platform for simulating hippocampal arrays based on ARM, providing the possibility of building a semi-physical simulation system suitable for hippocampal electrical stimulation experiments. Solutions to several key issues in the implementation process are proposed. A distributed computing method for the hippocampal array is attempted to be proposed, and a parallel computing hardware platform based on ARM is built for simulating epilepsy, thereby providing the possibility of embedded edge computing for epileptic focus localization. Solutions to several key issues in the implementation process are presented.

The remaining sections of this article are organized as follows. In Section 2, we introduce the hippocampal neuron array used in the semi-physical simulation platform and present the simulation results on a GPU platform. We analyze the advantages of this model in terms of simulation. In Section 3, we describe the structure of the semi-physical simulation platform and discuss how the parallel computing platform is utilized to enhance the overall computational power of the platform, providing a computational foundation for simulations conducted on the platform. Section 3 focuses on the algorithm simplification and task allocation mechanism for the hippocampal pyramidal neuron model suitable for the simulation platform. We optimize the floating-point computation process in the network, enabling the platform to overcome some of the difficulties associated with virtual simulation. Moving forward, in Section 4, we present the experimental validation and results analysis. The experiments aim to demonstrate the feasibility and effectiveness of the semi-physical simulation platform. Finally, in Section 5, we provide a summary of this article.

2. Computational Model of Hippocampal Neuron Array Coupled by Electric Fields

2.1. Unfolded Hippocampal Preparation

Hippocampal slices from the hippocampus of rats were prepared for use in electrophysiological experiments. CD1 mice of either sex from Charles River at postnatal day 10 (P10)–P20 were used. The unfolded hippocampus was prepared by following the surgical procedure described previously. A single hippocampus was dissected from the temporal lobe of the brain and unfolded by custom-made fire-polished glass pipette tools and a metal wire loop (Figure 1). The artificial CSF (aCSF) buffer contained the following concentrations (in mM): NaCl 124, KCl 3.75, KH_2PO_4 1.25, MgSO_4 2, NaHCO_3 26, dextrose 10, and CaCl_2 2. Additionally, 4-Aminopyridine (4-AP) was added to the normal aCSF at a final concentration of 100 μM [3,4]. During the experiment, an unfolded hippocampus was placed on top of the array with microelectrodes. Signals from the array were digitized by a DAP 5400a A/D system. Using the experimental setup described above, we conducted electrical stimulation experiments on the hippocampus and analyzed the potential data measured by electrodes, which allowed us to observe the coexistence of fast and slow traveling waves in the hippocampus. The in vitro environment of the hippocampal tissue was manipulated to observe the traveling wave state of the hippocampus under different conditions. Additional electrical stimulation experiments were performed to investigate the properties of hippocampal traveling waves in more detail [6].

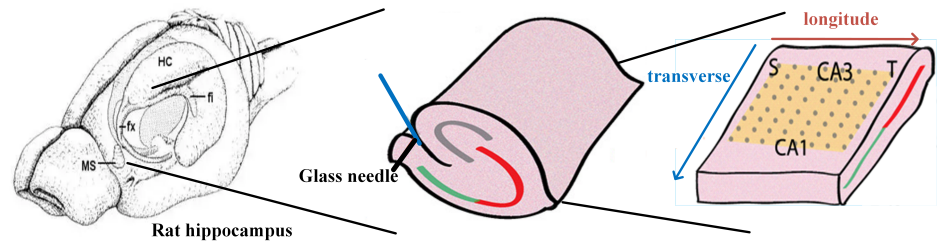


Figure 1. This figure depicting the process of preparing hippocampal slices was adapted from the figure shown in [4]. The left hippocampus was dissected from the septal-temporal lobe. A sharp glass needle was used to cut off the perforant path along the hippocampal fissure and the hippocampus was unfolded by pulling over the Dentate Gyrus (DG) to further flatten the folded curved structure.

2.2. Hippocampal Network Model Coupled by Electric Fields

A hippocampal network model neuronal firing network was constructed based on the above existing hippocampus structure. The object selected for simulation was a section of the rat hippocampus. The neuronal array simulates a region of approximately $3500 \mu\text{m}$ (X) \times $3000 \mu\text{m}$ (Y) \times $360 \mu\text{m}$ (Z) in the hippocampus, from the hippocampal septum to the temporal lobe for the longitudinal part of the neural network, and from area CA1 to area CA3 for the lateral part of the neural network (Figure 2A) [18]. The neuronal network is located in an approximate rectangular space with the x -axis direction representing the longitudinal direction and the y -axis direction representing the transverse direction, as shown in Figure 2 [22]. The z -axis represents the stacking of multiple layers of neurons (Figure 2), and to simplify the model, we introduced a stacking factor (SF) to replace the mutual influence between different layers of neuronal networks. Therefore, the object of our study is a single-layer 200 (X-axis) \times 200 (Y-axis) neuron matrix (Figure 2A). To accurately describe the pyramidal neuron array, the simulation has established a virtual coordinate system for the pyramidal neurons and represents each pyramidal neuron using XY-axis coordinates [23].

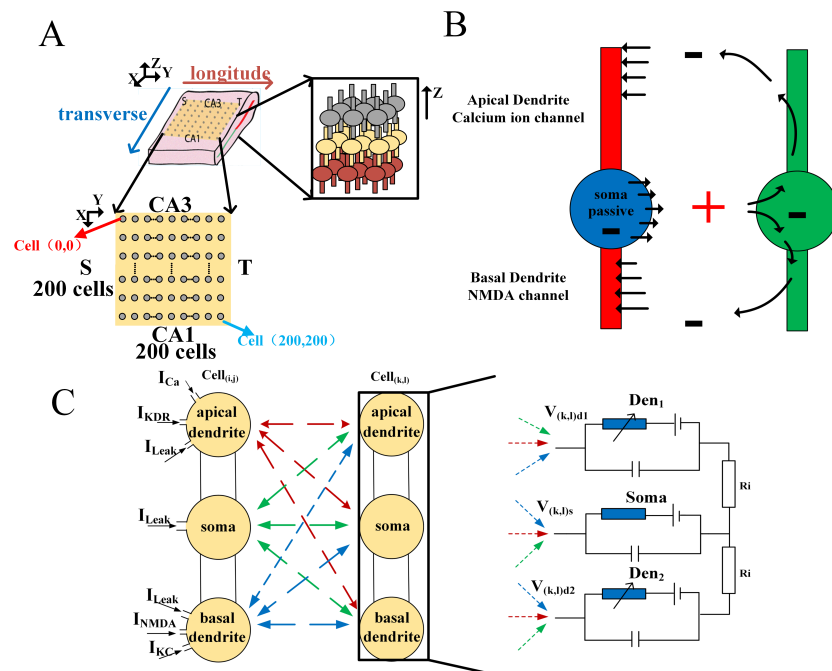


Figure 2. The basic structure of the hippocampal electrical stimulation simulation model includes: (A) neuron array architecture; (B) pyramidal neuron model; (C) electric field coupled model.

When an action potential occurs at the dendrite, current flows out the soma and in at the dendrites, creating a local current loop. This loop, in turn, depolarizes the adjacent

neuron that was originally at rest and modulates its electrical activity (Figure 2B). The conduction of electric fields is isotropic, but it is not realistic to fully achieve electric field conduction in all directions in the model. Studies have shown that the multidirectional electric field conduction of neurons mainly follows two vertical directions [22]. A hippocampal pyramidal neuronal network is a simulation network used as an alternative to in vitro experiments to facilitate a clearer and more convenient exploration of the working principles and pathological conditions of the hippocampus in the human brain [24]. Significant achievements have been made in the fields of human learning, memory, epilepsy, and Parkinson's disease through model simulations compared to in vitro experiments. The simulation model is based on the Hodgkin-Huxley (H-H) model [25], which closely resembles the physiological structure of neurons. The simulation model and electrophysiological model of the neuron are illustrated in Figure 2C. The relevant parameters of the neuron models are provided in Appendix A.

To verify that the neuronal spontaneous electric field is able to induce the slow oscillations found in vitro experiments, the model assumes bidirectional conduction of the neuronal electric field in both the X-axis and Y-axis directions and that the electric field exhibits bidirectional propagation in each axis. The calculation of the neuronal electric field follows exactly the laws of electrical correlation. Since each neuronal cell can be approximated as a point current source, according to Ohm's law and the potential calculation formula, the potential at any point in the electric field it generates can be determined (see Appendix B).

Compared to the models used in previous hippocampal simulation studies, the model employed in this paper incorporates a more complete three-compartment structure of neurons. While scaling up the neuron array, the model introduces bidirectional electric field conduction along two perpendicular directions. These improvements make the simulation results based on this model closer to the physiological electrical stimulation experiment of the hippocampus.

3. Embedded Parallel Computing Platform

3.1. The Basic Architecture of the Embedded Parallel Computing Platform

The embedded parallel computing architecture designed in this paper adopts an off-chip multi-core embedded structure. A single ARM is considered as the smallest Computing Unit (CU) and Routing Unit (RU), and an off-chip multi-core extension structure called Basic Extension Module (BEM) is proposed based on this CU [26]. The hardware modules mainly include the computation module and the routing module, where the computation module is the main structure for implementing the platform's main functions. The computation module houses a hippocampal pyramidal neuron array and an on-chip neural field connectivity network. Interactions between neural networks within the computation module are facilitated through a shared memory structure and the refreshing of neural field state queues. The expansion end of the computation module is equipped with various peripheral structures that support off-chip communication operations. The routing module is responsible for task allocation and data interaction among the computing cores, fulfilling the expansion requirements of the embedded platform. Shared Read-Only Memory (ROM) and Static Random-Access Memory (SRAM) connect all CUs and store model parameters and computation data. Through the cooperation of the above architecture, users can quickly build and freely expand an embedded parallel computing platform.

Figure 3 shows the design concept of the computation board. Considering the computational requirements of the computation board and the limitations of inter-board communication efficiency, each computation module integrates nine MCUs. These MCUs can be divided into two categories in terms of functionality: basic computing units responsible for computation and polling processors responsible for synchronization. There are eight MCUs as computing units, taking into account the computational power required for neuron calculations and communication efficiency requirements. Too many computing units would reduce the inter-board communication efficiency and not significantly im-

prove the computational efficiency. Each computing unit can execute tasks independently and store a certain number of pyramidal neurons during the simulation process. Usually, the pyramidal neurons of the neuron array are stored in each computing unit separately. To enable interactions among these pyramidal neurons and form a hippocampal neuron array, the platform establishes physical connections between the computation cores to facilitate data exchange. In terms of data communication, the computation module is directly connected to the polling processor via GPIO, achieving status acquisition and timing synchronization through high and low levels. The polling processor controls bus multiplexing, as multiple MCUs need to simultaneously occupy the data bus, and the polling processor can control the bus usage of the MCUs. Each computation board contains an off-chip shared memory SRAM, which is divided into private memory areas and a shared memory area. The private memory area is divided into eight blocks, which are mapped to each computing unit. The shared memory area can be accessed by all computing units and polling processors for storing discharge information sent to the computation cores.

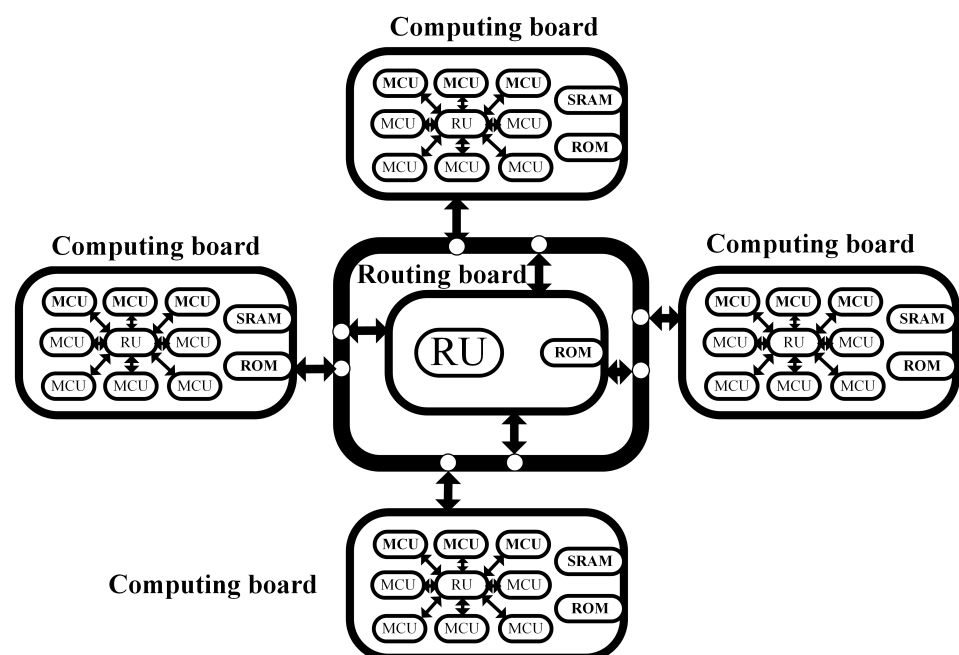


Figure 3. Modular and hierarchical off-chip multi-core computing architecture.

3.2. CUs Shared Resource Access and Communication Strategy

The structure of the hippocampal pyramidal neuron array system is relatively complex, and the computational tasks are heavy. The computational tasks need to be distributed to different computing cores for completion. In order to handle the complex computational tasks and distribute them efficiently, parallel computing platforms require collaboration and synchronization among the computing cores. Cooperative work among multiple computing cores is facilitated by utilizing a polling computing core in each computing board, which helps in coordinating the execution flow of the computing core programs.

The task flow in the computing core is shown in Figure 4 below. Upon powering on, the MCU waits for instructions from either the host computer or the routing board. Upon receiving instructions from the host computer, the MCU determines the neural models necessary for the computational task and reads the H-H model of the hippocampal pyramidal neuron array. Once the initialization is complete, the MCU initiates iterative computation. The initial membrane potential of the neurons and the parameters of channel conductance are read into the MCU and participate in real-time updates. Once the computation process is complete, the MCU generates a new set of variables that record the discharge state of all neurons associated with the MCU. The MCU can access the shared memory and upload the discharge state to the corresponding private memory area only

upon receiving the enable signal from the polling unit. Before obtaining polling permission, the MCU enters a self-loop state, waiting for the enable signal from the polling processor. This mechanism allows the polling processor to control the timing of all MCUs mounted on the computing board, enabling each MCU to occupy the FSMC data address bus and access the shared external SRAM. This helps in avoiding bus conflicts during shared memory access. The MCU's access to SRAM can be scheduled according to the timing control signals, enabling time-division multiplexing of data/address buses and enhancing program execution efficiency.

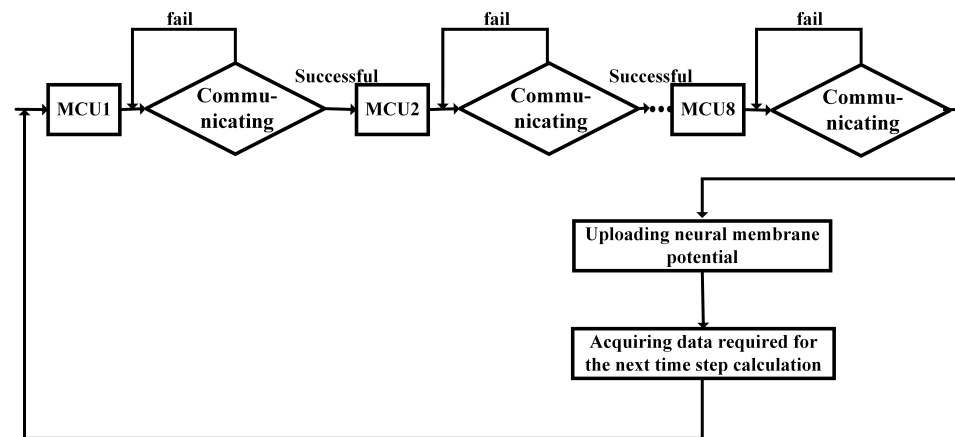


Figure 4. Communication flowchart of computing core under the control of routing polling unit.

This paper presents the design of an embedded multi-core platform that incorporates two communication channels: intra-board data exchange between computing modules and inter-board communication between the computing board and the routing board. The proposed approach utilizes bidirectional shared RAM, enabling high-speed inter-board communication at a transmission rate of up to 30 MB/s. The platform's communication speed has been enhanced compared to the inter-module communication speed of the STM32 chip series. With the STM32 series of microcontrollers, the transmission speed for inter-chip communication can vary between 200 kb/s and 20 MB/s, depending on the specific communication method and operating conditions. The inter-board communication speed of the platform also exhibits an advantage over other similar embedded computing platforms. This facilitates efficient communication between cores outside the computing modules. The flexible topology of the dual-port RAM enables the mapping strategy to accommodate various dimensions of hippocampal neural arrays. The dual-port RAM features bidirectional read and write capabilities, enabling simultaneous access to the same memory address from both directions of the bus, thereby enhancing the system's parallel efficiency. Each computing board incorporates off-chip shared memory SRAM, which is partitioned into private and shared memory regions. The private memory region is further divided into eight blocks, mapped to individual computing units. Each computing core possesses a private memory block with read and write permissions, while other cores have read-only access, preventing modification of the data within that memory block. The shared memory region is accessible to all computing units and the polling processor, serving as storage for discharge information from neurons dispatched to the computing cores. The dual-port SRAM is equipped with two read/write ports, enabling simultaneous read/write operations by different MCUs. The specific process is shown in Figure 5.

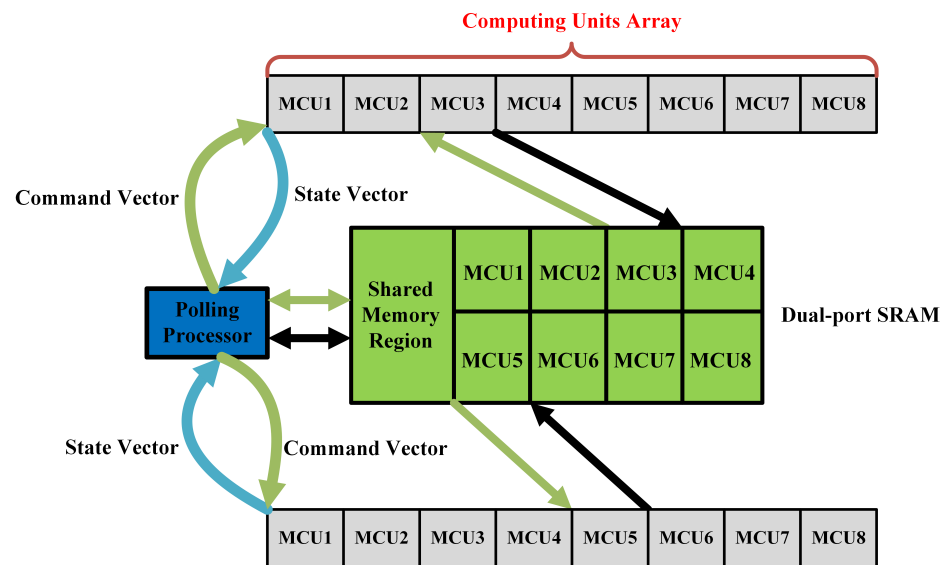


Figure 5. The working architecture of the computation board based on dual-port SRAM.

3.3. Physical Architecture and Task Implementation of Embedded Platforms

The STM32F407 series chip with the ARM architecture is selected as the CU and RU for the platform. The STM32F407 microcontroller has a built-in 1 MB ROM, divided into two areas: main flash and system memory. The main flash has a size of 1 MB for storing application code, while the system memory has a size of 64 KB for storing bootloader code, EEPROM emulator, and other system-related data. The access speed of the flash in the STM32F407 microcontroller depends on various factors such as access mode and operating frequency, with a maximum access speed of up to 30 MHz. In addition, the STM32F407 also features an Adaptive Real-Time (ART) accelerator, which utilizes prefetch and cache technologies to accelerate flash access speed and improve code execution efficiency. The STM32 has built-in interfaces for various protocols such as I2C, UART, and CAN. Based on the above architecture, a multi-core embedded parallel computing platform (MEPP) is designed. The MEPP consists of four computation boards and one routing board. The computation boards are responsible for storing neuron arrays and performing iterative operations on hippocampal neuron arrays. The routing board is mainly used for coordinating the scheduling of the four computation boards in the parallel computing system and performing data analysis and communication on the data uploaded by the computation boards. In addition, both the computation boards and the routing board include GPIO, USART, USB, and other expansion interfaces for communication with the host computer and real-time transmission of simulation data. Figures 6 and 7 show the actual picture of the simulation platform. In addition to the STM32 controller, the routing module and the computation module also include shared ROM, dual-port SRAM, expansion bus interface, and chip select programming terminals. Moreover, it has been calculated that the platform also has cost advantages. The entire platform utilized 36 STM32F407 chips, resulting in a total cost of around \$50. In contrast, a single Raspberry Pi typically costs between \$60 and \$70. More advanced embedded platforms and CPU-based platforms often have prices exceeding \$300.

On this embedded platform, numerous computational tasks related to neural networks have been accomplished [26]. Target recognition and tracking for the mobile robot have been achieved using Convolutional Neural Networks (CNN), leveraging the embedded platform and the front-end camera module. A Spiking Neural Network has been deployed on the platform to enable handwritten digit recognition, and a fully connected network based on the Leaky Integrate-and-Fire (LIF) neuron model has been designed. The experiments conducted validate the scalability and application potential of this parallel computing platform in large-scale neural networks [27]. However, the pyramidal neurons in the H-H model, which are closer to the physiological morphology of neurons compared

to the LIF model, involve more complex data. To enable real-time, semi-physical simulation on the embedded platform, relevant optimizations must be made to the platform and the pyramidal neuron model.

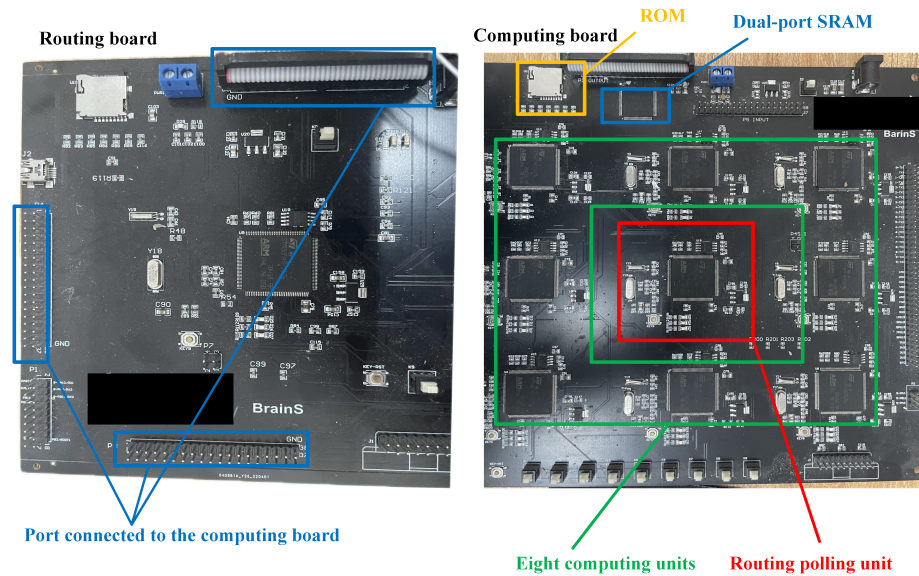


Figure 6. The physical map of routing board and computing board.

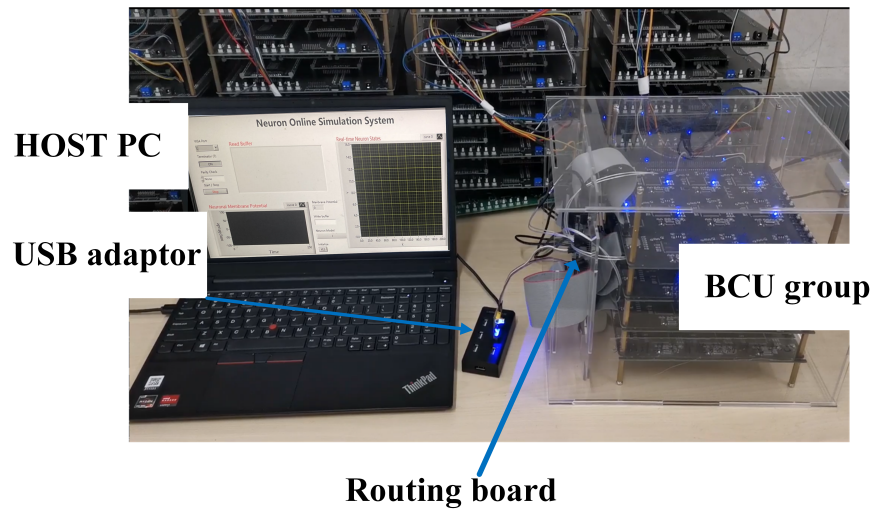


Figure 7. The actual architecture of an in-the-loop simulation platform.

4. Optimization of Hippocampal Network Model for Real-time Operation in Embedded Computing Platform

The hippocampal neural array is a novel simulated neural model that employs the Hodgkin-Huxley (H-H) model, which closely approximates the physiological state of pyramidal neurons, to simulate the states of pyramidal neurons within the hippocampus. A typical hippocampal array represents pyramidal neurons spanning from the temporal to septal regions of the hippocampus. The hippocampal model used for simulating epilepsy focuses primarily on the electric field propagation among pyramidal neurons, which differs from traditional synaptic and chemical transmission. Due to the influence of electric field propagation, the behavior of pyramidal neurons is no longer solely dictated by direct connections with neighboring layers, but rather determined collectively by all pyramidal neurons within the neural array. On a multi-core embedded parallel computing platform, the neural array needs to perform segmented computations and integrate the computed

results. The algorithm implemented on the routing board determines the location and movement speed of the current epileptic focus based on the real-time membrane potentials of pyramidal neurons within the array, transmitted by the computing cores. The forward computation process of the model involves extensive matrix multiplication and floating-point calculations. As the dimension and size of the model used in simulations increase, the computational complexity of the pyramidal neural array exhibits exponential growth. Conducting sequential computations on a single chip would result in substantial time costs. Furthermore, the limited storage space on the embedded platform requires significant storage resources to preserve weight parameters during each iteration step. Prior to integrating the model into the parallel computing platform, these two challenges must be addressed. To address these challenges, we propose mechanisms for dimensionality reduction initialization and model weight distribution for the hippocampal model, aiming to resolve the storage and computational limitations that restrict the application of the model.

4.1. Computation Reduction Initialization of Hippocampal Pyramidal Neuronal Networks

On a multi-core embedded hardware platform, there are several challenges in implementing distributed parallel computation for hippocampal models. When implementing hippocampal models on hardware platforms, the first consideration is the hardware storage resources [28]. The memory size of STM32F4 series chips is 192 KB, including 128 KB of RAM and 64 KB of CCRAM. In the computation of each pyramidal neuron layer, memory needs to be allocated for the input and output data of that layer, i.e., the initialization of the network layer.

To address the challenge of limited memory resources, we propose a method that combines dynamic memory allocation and dimensionality reduction initialization. Dynamic memory allocation is a technique used for allocating and managing computer memory resources. It allocates the required memory from a memory pool when executing a memory application. After processing the data, the memory resources are released and reclaimed. However, due to the non-continuous addresses of the two random access memory (RAM) modules, the actual RAM size that supports dynamic memory allocation is limited to 128 KB. Therefore, the amount of data allocated through dynamic memory allocation cannot exceed this limit at any time.

Since individual network layers typically involve a large amount of data, relying solely on dynamic memory allocation is not sufficient. To address this issue, we propose a dimensionality reduction initialization method. Typically, the physiological structure of the hippocampal array is a curved spiral, which is often flattened into a rectangular slice in experiments, and the model simulates a pyramidal neuronal array within a cuboid between the temporal lobe and the septum. The electric field propagation in the hippocampal pyramidal neuronal network is represented as three-dimensional conduction, where the electric field can propagate in any direction within the cuboid. Through dimensionality reduction initialization, we allocate memory only for a two-dimensional matrix (i.e., a single feature map) when initializing an individual network layer. Once the computation within a single stride is completed, we transfer the data to dual-port RAM, erase the data of the current stride, and write the data for the next stride. If the amount of data generated within a single iteration stride is still too large, we can further reduce the dimensionality of the matrix allocation by allocating memory for multiple two-dimensional arrays, avoiding the creation of large three-dimensional arrays and reducing the amount of data transmitted during the communication process. This approach minimizes the impact on the hippocampal network while optimizing the utilization of platform memory.

4.2. The Memory Allocation Mechanism in the Hippocampal Pyramidal Neuron Network

For large-scale neural networks like the hippocampal pyramidal neuron network, the storage consumption of membrane potential parameters typically increases exponentially with network size. In order to achieve a trade-off between network accuracy and computational cost, the hippocampal neuron network adopts two strategies: potential value

quantization and efficient memory mapping, to reduce the computational, communication, and storage overhead.

During the iteration process of the hippocampal array, there are a significant number of floating-point operations. Although the STM32F4 series microprocessors are compatible with floating-point operation modules, the transmission cost and memory usage of floating-point numbers are much higher than those of integer numbers. When using random inputs, inference can be achieved using low-bit-width weights without significant loss in discrimination accuracy. Similarly, in the process of discerning the location of wave sources, reducing the data bit width can be employed by using binary weights instead of full-precision floating-point weights. To maintain computational accuracy, floating-point parameters are still used in the calculation board, but the number of bits for floating-point numbers is reduced from 32 bits on the CPU platform to 8 bits, reducing the data overhead during communication and memory processes. After the computation is completed, the calculation results of the pyramidal neurons are quantized by the calculation core. The quantization function used in this paper can be represented as:

$$\omega_b = \text{floor}\left(\frac{\omega - \omega_{\min}}{\omega_{\max} - \omega_{\min}}(2^k - 1)\right), \quad (1)$$

where ω represents the initial continuous weight, and ω_b represents the quantized weight. This formula allows the mapping of the neuron's weights to k-bit binary numbers in Figure 8. A smaller quantized weight bit width has a greater impact on the accuracy of position determination. The calculation board transmits these binary weight values to the routing board, which locates the current fast and slow wave sources within the array based on the binary weight values of the neuron array.

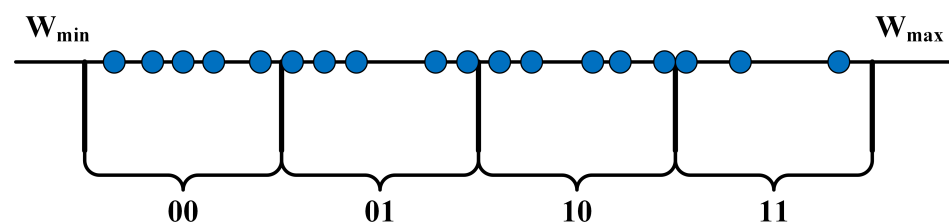


Figure 8. The distribution of compressed weight mapping intervals.

Another key aspect of implementing the CNN model on a multi-core embedded hardware platform is how to effectively map the neuron array to multiple computing units. We propose a hippocampal array distribution mechanism based on the computing and communication capabilities of the calculation cores. This mechanism includes horizontal layering, bidirectional conduction, and weight calculation. Through this approach shown in Figure 9, the entire network can be gradually dispersed.

As a parallel network model, the hippocampal neuron array itself does not have a hierarchical structure, and the pyramidal neurons are influenced by all the neurons within the array. At the same time, all neurons complete one time-step iteration in different computing cores. Due to the fixed computational tasks of each computing unit, when executing the hippocampal network mapping, issues such as inter-module data dependencies and load balancing need to be considered. All neurons in the four calculation boards complete the computation of the first iteration step based on the initial conditions provided by the routing board, and the membrane potential weight conversion is performed within the calculation board. The calculation board stores the current time-step neuron potentials in dual-port SRAM, transfers the weights to the routing board for wave source determination, and waits in the routing board for permission from the host computer to transmit the computation results. This process is looped for the next time step.

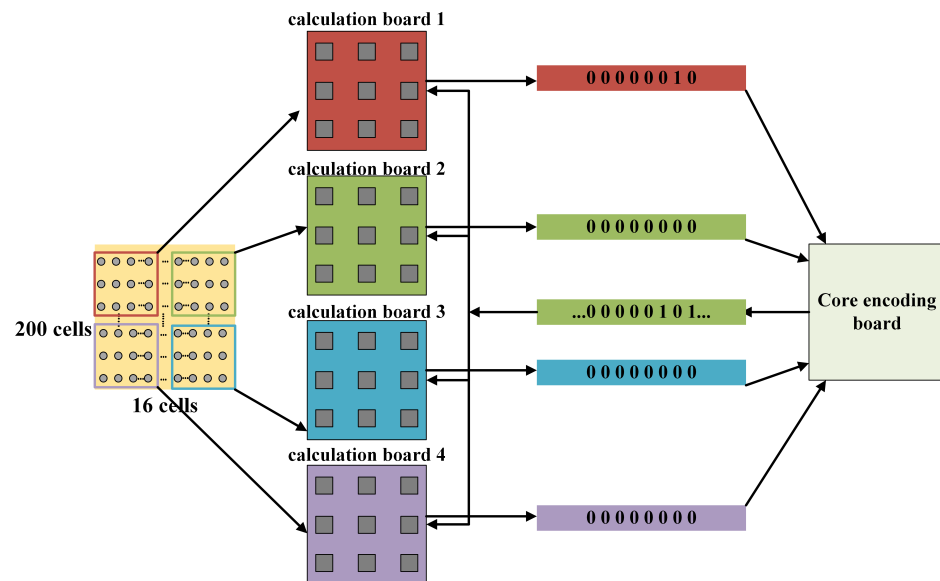


Figure 9. The functional architecture and memory allocation of the experimental platform.

5. Results

In the experiment, real-time electrical stimulation simulation of hippocampal arrays was successfully conducted on an embedded platform, and real-time display of simulation results was achieved on the host computer (Figure 10). The membrane potential of neurons was displayed in the lower-left corner, while the discharge status of the current neuron array was displayed in the upper-right corner. By observing the discharge patterns, the current wave source situation within the array can be inferred. The real-time display of membrane potentials in different compartments of neurons revealed slow wave oscillations in the apical dendritic membrane potential and fast wave oscillations in the basal dendritic membrane potential. By modifying the conductance of relevant neuron channels, similar physiological changes in the propagation of fast and slow waves were observed, consistent with previous model simulation results. Furthermore, by displaying the discharge patterns of the neuron array on the host computer, the researchers were able to observe the propagation of fast waves within the neuron array, as depicted in Figure 10. This demonstrated the correctness and rationality of the semi-physical simulation platform throughout the entire simulation process.

The membrane potentials obtained from *in silico* simulations will be imported into data analysis software for further analysis. By comparing the results with those obtained from previous *in silico* simulations, the following results, as shown in Figures 11–14, were obtained. In Figure 11, the fast spike and slow wave obtained from the in-loop simulation showed similarities with the results obtained from *in silico* simulations. The traveling waves obtained from platform simulation have the same waveform and similar traveling wave speed. Figure 12 illustrates the changes in the excitability of fast and slow waves and the coupling strength of the electric field within the neuron array after adjusting the model stacking coefficient. The in-loop simulation platform, *in silico* platform, and *in vitro* experimental results show the same trend of change which proved the reliability of simulation platform. Figure 13 illustrated the comparison of parameters related to fast spikes and slow waves, showing that the in-loop simulation results were consistent with the *in silico* simulation results. Figure 14 illustrated the comparison of parameters related to fast spikes and slow waves, showing that the in-loop simulation results were consistent with the *in vitro* experiment results. Within the neuronal array, the propagation of fast spikes and slow waves could be observed, as shown in Figures 15 and 16. The simulation revealed that the source of fast waves was not located at a fixed position. Over the 300 ms of simulated observation, two instances of fast wave propagation with different source positions within the array were detected. On the other hand, during the simulation of slow

waves, the position of the slow wave source remained relatively constant, and the velocity was significantly slower compared to fast waves. The obtained results are consistent with the simulated results within the computer.

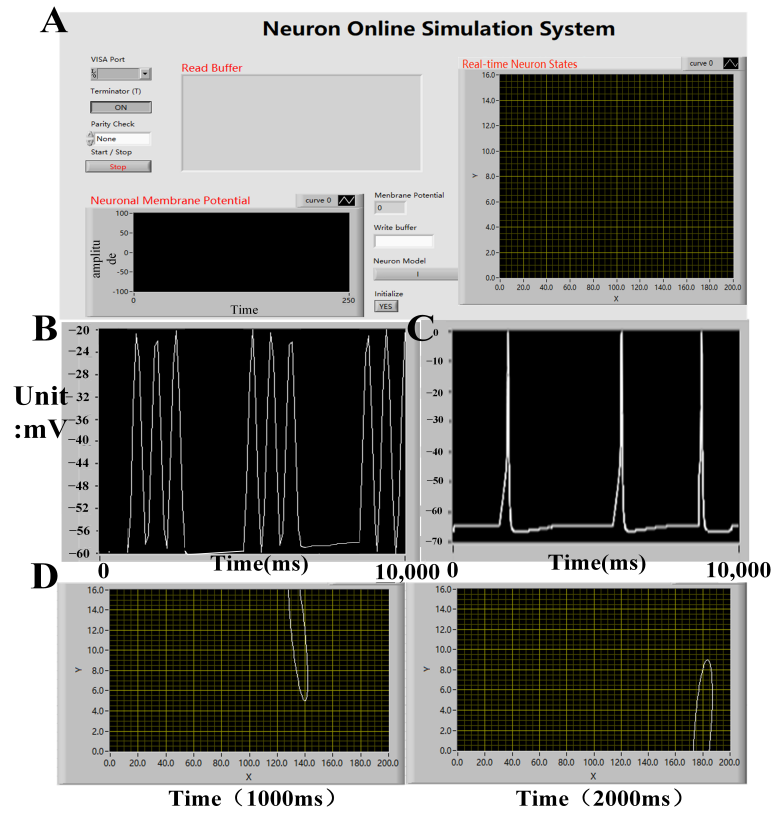


Figure 10. (A) The GUI design of the upper computer platform on the in silico simulation platform allows for the control of simulation system parameters, real-time membrane potential waveforms of neurons, and the propagation status of waves within the neuronal array. (B) The holder computer displays the fast and slow waveforms of neurons (100, 8) in neuron array. (C) The holder computer displays the fast and slow waveforms of neurons. (D) The propagation of fast waves within the neuronal array.

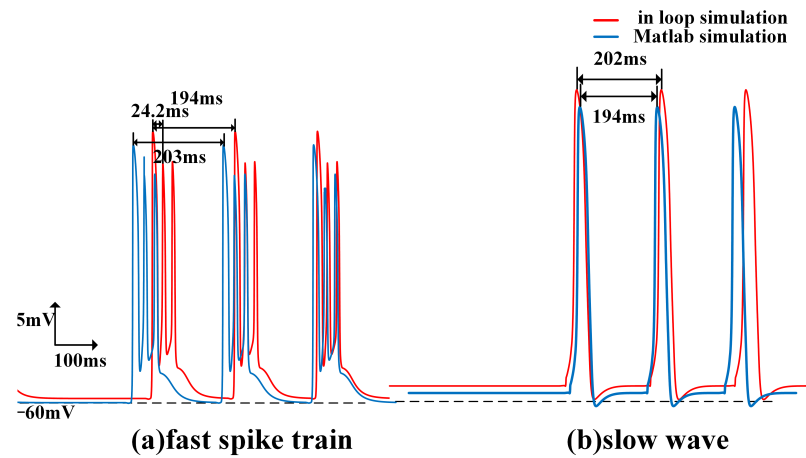


Figure 11. Fast and slow waves of cell (100, 8) in the hippocampus array obtained through in loop and Matlab simulation, with slow waves on the right and fast spikes on the left.

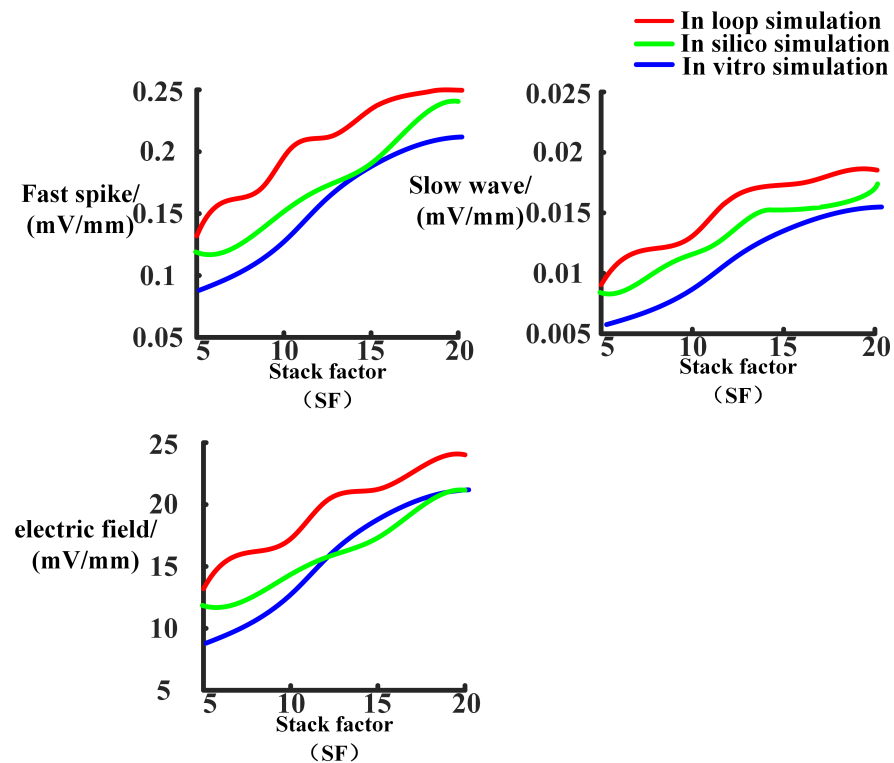


Figure 12. Changing the stacking factor of the model, the traveling wave speed and hippocampal electric field strength were tested.

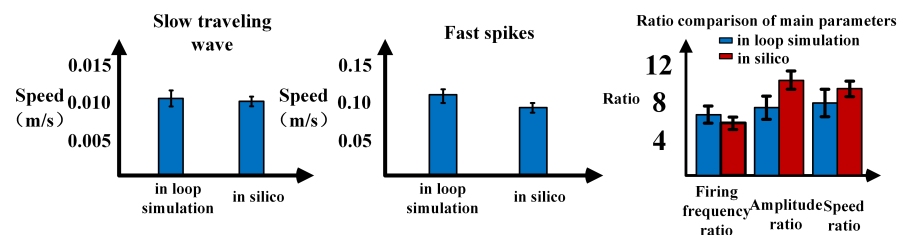


Figure 13. Comparison of traveling wave data between in-loop simulation and computer simulation.

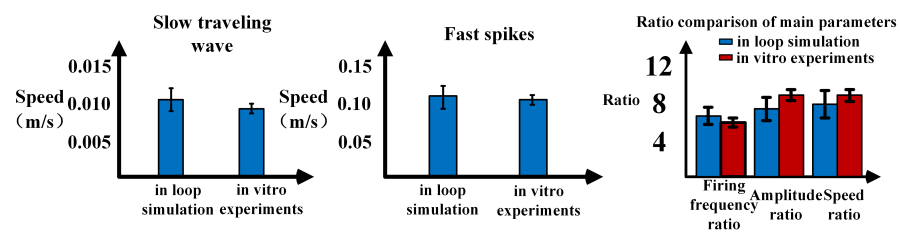


Figure 14. Comparison of traveling wave data between in-loop simulation and Matlab simulation.

When compared to in silico simulations, the in-loop simulation exhibits an increase in variance in the measured data and a decrease in measurement accuracy due to parameter and model simplifications (Figure 13). Compared with the results obtained from the same scale model simulation conducted on the computer, the average error of fast and slow wave velocities is 4.6%, and the variance of wave velocity measurement has increased by 8.2%. However, considering the optimization of computational power and memory achieved through these simplifications, the minor sacrifice in accuracy is acceptable. Regarding the propagation of waves, the simplified model scales result in changes to the observed wave propagation process. The phenomenon of fast wave source movement within a single cycle is observed only once. Nevertheless, the in silico simulation on the embedded platform still adequately reflects the conduction process of traveling waves within the hippocampus.

After verifying the accuracy of the simulation results within the loop, the real-time performance and feasibility of the loop simulation platform are tested using various computations. A hippocampal pyramidal neuron array model with different input sizes was implemented on this hardware platform. The hippocampal models with varying input sizes are shown in Figure 17A. Different experiments were designed to verify the effectiveness of dimensionality reduction initialization and the hippocampal distribution mapping mechanism during the implementation process. Finally, an analysis of the platform's runtime, resource consumption, and power usage was conducted.

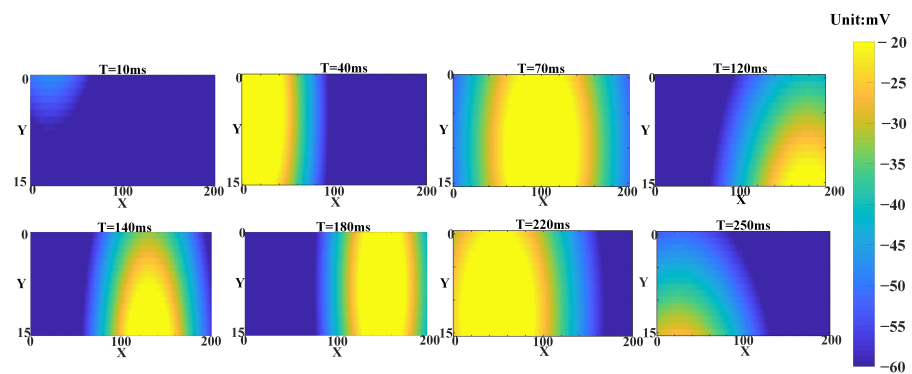


Figure 15. The membrane potentials of hippocampal neurons obtained from the in-loop simulation on the embedded platform were analyzed. The membrane potentials within the hippocampal array were visualized as a heatmap, revealing the propagation process of fast waves within the neurons and the movement of wave sources during conduction.

The effectiveness of the method was validated by comparing the memory usage of the models before and after dimensionality reduction initialization. Figure 8 indicates the memory required by each model during the computational process when the input model size is a 200×16 three-dimensional array and a 200×16 three-dimensional array. For an input neuron matrix array size of 200×200 , the memory required for data from 100 H-H neurons could be accommodated by a single chip. However, when the input pyramidal neuron matrix size is 200×200 , the memory required for a single network layer exceeds the platform's maximum memory capacity. Therefore, the experiments primarily focused on the 200×16 pyramidal neuron array.

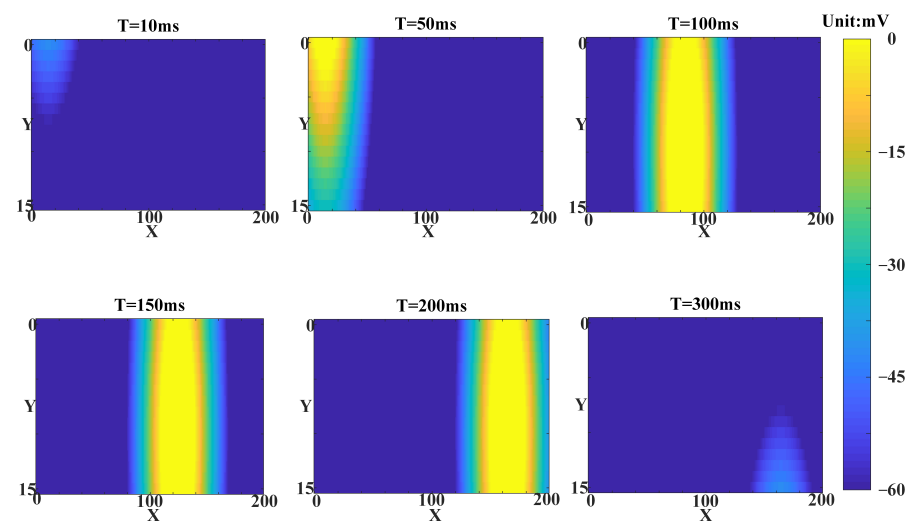


Figure 16. The simulation utilizes a heatmap to reproduce a single conduction process of slow waves within the hippocampal array. The source position of slow wave conduction remains relatively unchanged and has a higher amplitude compared to fast waves.

Different sizes of hippocampal neural array models were implemented on an embedded platform without considering real-time performance. The algorithms used in this article are mostly linear operations on two-dimensional matrices. The time complexity of the algorithm is $O(n^2)$ and the spatial complexity of the algorithm is also $O(n^2)$. To evaluate the platform's real-time performance, the neural network dynamics were simulated within a 100-second timescale, with a simulation time step of 1 ms and a total of 10,000 iterations of neuron computations. Figure 17 illustrates the time consumption. When the model's input size is $200 \times 16 \times 3$ H-H pyramidal neurons, the model's runtime reaches 154 s, exceeding the required 100 s for real-time performance. The majority of the network computation time is occupied by floating-point calculations and communications. After optimizing the floating-point calculations, the system's real-time performance was further tested. As the number of H-H neurons increased, the simulation time on the platform also increased. When the number of H-H pyramidal neurons in the platform reached 3200, the simulation time reached 94 s, indicating that the model simulation has some level of real-time performance after lightweight improvements. Compared to two-dimensional matrix simulation, the simulation time of the reduced-dimensional three-dimensional neural network array exhibits a smaller increase.

For the embedded computing platform, memory resource consumption primarily includes SRAM and FLASH. When the input matrix size is 200×16 , the platform's memory requirements can be met. Therefore, dimensionality reduction initialization was performed on the hippocampal pyramidal neuron array to reduce the input size to a 200×16 two-dimensional matrix. The computation of the three-dimensional neuron array was transformed into a two-dimensional matrix computation using stacking coefficients. Figure 17C demonstrates that the memory occupied by the initialization of a single network layer in the 200×16 pyramidal neuron model, after dimensionality reduction initialization, is effectively limited within the chip's memory size. Each computing core, consisting of 100 H-H pyramidal neurons, occupies 7 kB of on-chip SRAM space and 24.4 kB of on-chip FLASH space. With the platform supporting 512 KB FLASH and 128 KB SRAM, the on-chip memory resources can accommodate the required size of the neuron cluster for this model computation. The analysis results demonstrate the effectiveness of dynamic memory allocation and dimensionality reduction initialization methods. This approach can fully address the issue of insufficient memory.

Through the aforementioned experimental tests, we have demonstrated the computational feasibility of the optimized hippocampal pyramidal neuron model on the parallel computing platform. We have obtained corresponding results through computations, and the discharge waveforms of the pyramidal neurons and the discharge point map of neurons within the array can be observed through the host computer platform. The routing board can determine the location of the current wave source based on the weights. It is evident that the pyramidal neuron array can perform its intended tasks excellently on the embedded parallel platform.

The time consumed during a single computation process by the MCU computing core includes both the polling communication time and the computation time. Under the assumption of constant computing power, a lower polling time indicates a smaller proportion of the computation cycle dedicated to polling, implying that more resources are allocated to the computation task, thereby improving computational efficiency. The Percentage of Polling Time (PCP) is defined as:

$$PCP = \frac{T_{rout}}{T_{cir}} \times 100\%, \quad (2)$$

where T_{rout} represents the polling time and T_{cir} represents the computation cycle time. By varying the number of H-H neurons within the computing unit, the polling time during the iterative computation process is measured, as shown in Figure 17D. It can be observed that when the number of neurons is small, a significant portion of system resources is allocated to polling communication. As the number of H-H neurons reaches around 100,

the PCP approaches 50%, indicating that half of the resources are dedicated to computation. The curve demonstrates that deploying a large number of neurons in the MCU can reduce the impact of polling time on the experiment. However, an excessive number of neurons may compromise system real-time performance. Taking various factors into consideration, setting 100 H-H neurons within a single MCU is acceptable in terms of platform efficiency and system real-time performance.

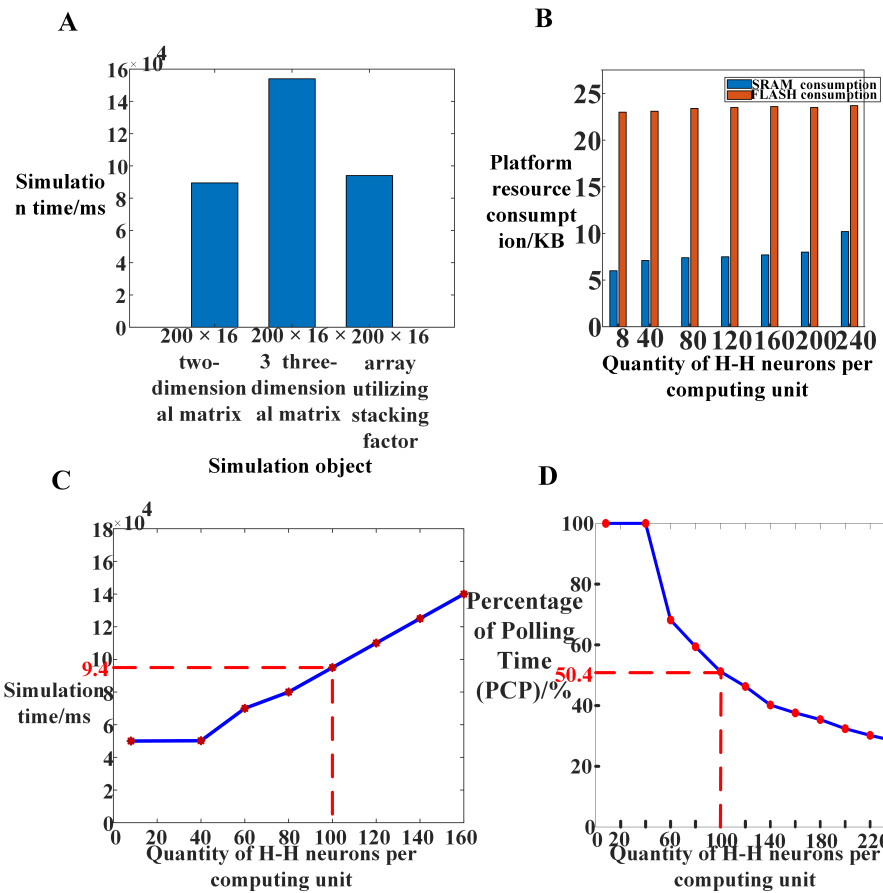


Figure 17. Different methods were used to measure the real-time, synchronous, and other performance indicators of hippocampal electrical stimulation simulation on the loop simulation platform. (A) The actual time consumed by different simulation objects during a 100 ms simulation process in the loop simulation platform. (B) The memory utilization rate of the platform when accommodating different numbers of H-H model neurons in the computing core. (C) The actual time consumed by the system during a 100 ms simulation process when accommodating different numbers of H-H model neurons in the computing core. (D) The percentage of polling time (PCP) of the system when accommodating different numbers of H-H model neurons in the computing core.

Table 1 provides the power consumption of the hippocampal pyramidal neuron array platform before and after optimization in the experiments. The platform current is measured from the system’s power supply, and the consumption of storage components such as SRAM can be obtained through burning software. Tables 2–4 compares the performance of embedded platform with computer simulation. The hippocampal traveling wave velocity obtained from embedded platform and CPU simulation were compared and the error between the two simulations was obtained. By comparison, it can be seen that the error of the embedded platform has increased by 1.2% compared to CPU simulation. CPU simulation speed is faster than embedded platforms. The energy consumption of CPU during simulation is much higher than that of embedded platforms. Notably, when the number of computational bits is increased, the simulation error between CPU simulation and embedded simulation becomes more comparable. However, increasing the number

of computing bits on embedded platforms means a decrease in real-time simulation performance. After increasing the number of simulation bits to eight, the simulation time of the platform exceeds 100 s, and the platform no longer has real-time performance. From the data in the table, it can be inferred that after optimization such as dimensionality reduction and weight parameter compression, the hippocampal pyramidal neuron network can perform real-time simulation operations at a lower power consumption and relatively high speed within the acceptable range of a 1.2% decrease in accuracy. The platform accomplished the simulation computing task with a power consumption of 486.6 mW. It exhibits a significant advantage in power consumption while the power consumption of the Raspberry Pi Zero typically ranges from 1.0 to 1.5 watts and the Raspberry Pi 4B has relatively higher power consumption, typically ranging from 3.5 to 7.0 watts. This is one of the significant advantages of the platform in specific application scenarios.

Table 1. Power consumption of the embedded platform MEPP.

Measurement Object	Working Current	Single Chip Power Consumption
Calculation unit	19.3 mA	67.23 mW
Embedded platform	70.3 mA	486.6 mW

Table 2. Comparison between embedded simulation platform and CPU simulation under 4 computational bits.

Platform	Execution Time ¹	Accuracy ²	Number of Bits	PCP	Power Consumption ³
Embedded platform	94 s	1.4%	4	50.4%	486.6 mW
CPU ⁴	71 s	\	4	62.8%	6.3 W

¹ The execution time refers to the actual duration required to complete a 100-second simulation within the model. ² The accuracy of the table is the error in comparing the simulation results of fast and slow traveling wave velocities with the results of in silico simulation [23]. ³ The energy consumption calculation method for embedded platforms has been mentioned in this article. For the energy consumption calculation of CPU simulation process, only the energy consumption of CPU was measured using third-party software, excluding other computer devices. The measured power is the average power within 100 s of the actual time. ⁴ The CPU platform we selected was the Intel(R) Core(TM) i5-11400.

Table 3. Comparison between embedded simulation platform and CPU simulation under 8 computational bits.

Platform	Execution Time	Accuracy	Number of Bits	PCP	Power Consumption
Embedded platform	110 s	0.9%	8	42.4%	874.3 mW
CPU	81 s	\	8	52.8%	12.5 W

Table 4. Comparison between embedded simulation platform and CPU simulation under 16 computational bits.

Platform	Execution Time	Accuracy	Number of Bits	PCP	Power Consumption
Embedded platform	122 s	0.6%	16	22.4%	1.3 W
CPU	88 s	\	16	34.2%	19.2 W

6. Discussion and Conclusions

The loop simulation system provides a promising platform for simulating hippocampal electrical stimulation experiments. In this study, an embedded parallel computing platform was used to perform loop simulations of hippocampal electrical stimulation, and the simulation performance of the loop simulation system was evaluated. The experimental results demonstrate that the neural network parallel computing hardware platform can realize the hippocampal pyramidal neuron array. It offers advantages such

as low power consumption, scalability, and low cost. Additionally, the effectiveness of dimensionality reduction initialization and pyramidal neuron distribution mechanism in achieving real-time simulation of epileptic neural networks has been validated. In future work, we will consider modifying the communication method and control chip of the board. Faster data transmission and higher-end chips can better guarantee real-time performance. Building upon the current architecture, we can replace the relatively low-frequency and computing-power STM32F407 series chips currently used on the platform with more powerful chips to achieve more efficient computational capabilities.

The real-time of model computations is one of the core issues in convolutional neural network hardware acceleration. Real-time performance is mainly limited by computational capabilities and inter-core communication latency. In this experiment, we demonstrated the effectiveness of high-speed communication through shared RAM. However, the performance of the hardware platform has not been fully utilized, and future work needs to discuss methods for achieving higher computational capabilities. Introducing more powerful cores and larger chip external storage space under the current architecture can accommodate diverse inter-chip communication methods. Additionally, the platform currently only supports operations on large-scale hippocampal arrays, limiting its application scope. However, other physiological neural networks can perform distributed computations based on the structure of our platform. In the future, we will investigate the performance of other physiological neural networks on the platform. In this experiment, we demonstrated the effectiveness of high-speed communication through shared RAM. However, the performance of the hardware platform has not been fully utilized, and future work needs to explore methods for achieving higher computational capabilities. Under the current architecture, introducing more powerful cores and larger external storage space can accommodate more diverse external communication methods. In future research, we will continue to optimize our platform and adapted networks in two aspects: reducing network size and compressing weight parameters to achieve more diverse parallel computational capabilities.

In addition, we plan to further expand the application scenarios of the platform by adding hardware resources such as EEG monitoring and camera modules to enable real-time monitoring of epileptic patients. Furthermore, the current platform can only support operations on large-scale hippocampal arrays, limiting its application scope. However, other physiological neural networks can be distributed computations based on the structure of our platform. In future research, we will explore the performance of other physiological neural networks on the platform. Based on the hippocampal pyramidal neuron experiment, we also conducted experiments on more complex Rubin-Terman physiological neural networks on the platform. In addition to designing multi-core systems, we have also designed a lightweight physiological neural network for multi-core systems to optimize the model deployment process and make network design more suitable for the hardware. In future research, we will continue to optimize our platform and adapted networks in two aspects: reducing network size and compressing weight parameters to achieve more diverse parallel computational capabilities.

Author Contributions: Conceptualization, X.W.; methodology, X.W.; software, Z.R.; validation, X.W., Z.R. and M.L.; formal analysis, M.L.; investigation, X.W.; resources, X.W.; data curation, X.W.; writing—original draft preparation, Z.R.; writing—review and editing, X.W., Z.R., M.L. and S.C.; visualization, Z.R.; supervision, X.W.; project administration, X.W. and S.C.; funding acquisition, X.W. and S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant 62171312, 62271348, 62303345, the Tianjin Municipal Education Commission scientific research project under grant 2020KJ114 and Postdoctoral Fellowship Program of CPSF (No. GZB20230513).

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

The cell membrane model used in the model is the Hodgkin-Huxley model, which is closer to the physiological, electrochemical properties of real neurons. Some fixed conductance and potential in the model are shown in Tables A1 and A2.

Table A1. The electronic parameters in the computational model.

Parameter	Symbol	Value
Somatic membrane resistance	R_{ms}	680 Ω/cm^2
Dendritic membrane resistance	$R_{md1}R_{md2}$	780 Ω/cm^2
Intracellular resistance	R_i	530 Ω/cm^2
Extracellular resistance	R_e	300 Ω/cm^2
Dendrites membrane capacitance	C_{md}	1 $\mu F/cm^2$
Somatic membrane capacitance	C_{ms}	1 $\mu F/cm^2$

Table A2. Some fixed conductance and potentials in the equation [23].

Parameter	Symbol	Value
NMDA channel conductance	g_{NMDA}	6.5 ms/cm ²
Potassium ion channel conductance	g_{KDR}	200 ms/cm ²
Somatic membrane leak conductance	g_{msLeak}	1.4706 ms/cm ²
Dendritic membrane leak conductance	g_{mdLeak}	0.0292 ms/cm ²
Resting potential of NMDA channels	E_{NMDA}	0 V
Resting potential of potassium ion channel conductance	E_K	-60 mV
Resting potential of leaking channel conductance	E_{Leak}	-58 mV
Resting potential of calcium ion channels	E_{Ca}	10 mV

Appendix B

The membrane potential calculation formula for the ion channels in the Hodgkin-Huxley (H-H) model used in this study is described as follows [29]:

$$\frac{dV_{md1}}{dt} = \frac{1}{C_{md}}(-I_{d1_NMDA} - I_{s_d1} - I_{d1_Leak} - I_{d1_stim} - I_{d1_d2} - I_{KDR}), \quad (A1)$$

$$\frac{dV_{ms}}{dt} = \frac{1}{C_{ms}}(-I_{d1_s} - I_{s_Leak} - I_{s_stim} - I_{d2_s}), \quad (A2)$$

$$\frac{dV_{md2}}{dt} = \frac{1}{C_{md}}(-I_{d2_Ca} - I_{d2_KC} - I_{d2_Leak} - I_{d2_stim} - I_{s_d2} - I_{d2_KAHP}), \quad (A3)$$

$$B = \frac{1}{1 + \exp(-k_b V_{md}) \frac{Mg^{2+}}{M_0}}, \quad (A4)$$

$$\frac{d\eta}{dt} = \alpha - \eta(\alpha + \beta), \quad (A5)$$

The gating formulas for NMDA channel are shown in the following equations [3].

$$I_{NDMA} = -g_{NMDA}B(V_{md1} - E_{NMDA}), \quad (A6)$$

$$B = \frac{1}{1 + \exp(-k_b V_{md}) \frac{Mg^{2+}}{M_0}}, \quad (A7)$$

The gating formulas for calcium are shown in the following equations [3].

$$m_{Ca} = \frac{m_{CaH}}{\tau_{mCaH}}, \quad (A8)$$

$$h_{Ca} = \frac{h_{CaH}}{\tau_{hCaH}}, \quad (A9)$$

$$I_{Ca} = -g_{m_Ca} m_{Ca}^2 h_{Ca} (V_{md2} - E_{Ca}), \quad (A10)$$

References

1. Terman, D.; Rubin, J.E.; Yew, A.C.; Wilson, C.J. Activity patterns in a model for the subthalamopallidal network of the basal ganglia. *J. Neurosci.* **2002**, *7*, 2963–2976. [[CrossRef](#)] [[PubMed](#)]
2. Follett, K.A.; Weaver, F.M.; Stern, M.; Hur, K.; Harris, C.L.; Luo, P.; Marks, W.J., Jr.; Rothlind, J.; Sagher, O.; Moy, C.; et al. Pallidal versus Subthalamic Deep-Brain Stimulation for Parkinson's Disease. *N. Engl. J. Med.* **2010**, *362*, 2077–2091. [[CrossRef](#)]
3. Chiang, C.C.; Wei, X.; Ananthakrishnan, A.K.; Shivacharan, R.S.; Gonzalez-Reyes, L.E.; Zhang, M.; Durand, D.M. Slow moving neural source in the epileptic hippocampus can mimic progression of human seizures. *Sci. Rep.* **2018**, *8*, 1564. [[CrossRef](#)]
4. Zhang, M.M.; Durand, D.M. Propagation of epileptiform activity in the hippocampus can be driven by non-synaptic mechanisms. In Proceedings of the 6th International IEEE EMBS Conference on Neural Engineering (NER), San Diego, CA, USA, 6–8 November 2013; pp. 790–793
5. Ghai, R.S.; Bikson, M.; Durand, D.M. Effects of applied electric fields on low-calcium epileptiform activity in the CA1 region of rat hippocampal slices. *J. Neurophysiol.* **2000**, *84*, 274–280. [[CrossRef](#)] [[PubMed](#)]
6. Qiu, C.; Shivacharan, R.S.; Zhang, M.M.; Durand, D.M. Can Neural Activity Propagate by Endogenous Electrical Field? *J. Neurosci.* **2015**, *35*, 15800–15811. [[CrossRef](#)] [[PubMed](#)]
7. Diba, K. Hippocampal sharp-wave ripples in cognitive map maintenance versus episodic simulation. *Neuron* **2021**, *109*, 3071–3074. [[CrossRef](#)] [[PubMed](#)]
8. Wu, Y.X.; Chen, Z. Computational models for state-dependent traveling waves in hippocampal formation. *BioRxiv* **2023**. [[CrossRef](#)]
9. Lemaire, M.; Sicard, P.; Bélanger, J. Prototyping and Testing Power Electronics Systems using Controller Hardware-In-the-Loop (HIL) and Power Hardware-In-the-Loop (PHIL) Simulations. In Proceedings of the 12th IEEE Vehicle Power and Propulsion Conference (VPPC), Montreal, QC, Canada, 19–22 October 2015.
10. Zhang, Y.; Shi, C.; Dai, H.Y. Design for Hardware in-the-Loop Real-Time Simulation Test of Combined Seeker. In Proceedings of the 9th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 10–11 December 2016; pp. 74–77.
11. Li, X.W.; Huang, X.S.; Sun, J.X. The Design of Strapdown TV Seeker Hardware-in-the-loop Simulation System. In Proceedings of the International Conference on Modeling, Beijing, China, 26–27 December 2009; pp. 59–62.
12. Song, H.H.; Xie, Y.M.; Cai, W.Y. Study on Hardware-in-the-loop-simulation of hydroturbine governing system. In Proceedings of the International Conference on Quantum, Nano and Micro Technologies, Beijing, China, 1–3 September 2009; pp. 59–62.
13. Brandman, D.M.; Burkhart, M.C.; Kelemen, J.; Franco, B.; Harrison, M.T.; Hochberg, L.R. Robust Closed-Loop Control of a Cursor in a Person with Tetraplegia using Gaussian Process Regression. *Neural Comput.* **2018**, *30*, 2986–3008. [[CrossRef](#)]
14. Ezzyat, Y.; Wanda, P.A.; Levy, D.F.; Kadel, A.; Aka, A.; Pedisich, I.; Sperling, M.R.; Sharan, A.D.; Davis, K.A.; Worrell, G.A.; et al. Closed-loop stimulation of temporal cortex rescues functional networks and improves memory. *Nat. Commun.* **2018**, *9*, 2041–2052. [[CrossRef](#)]
15. Eliasmith, C.; Stewart, T.C.; Choo, X.; Bekolay, T.; DeWolf, T.; Tang, Y.; Rasmussen, D. A Large-Scale Model of the Functioning Brain. *Science* **2012**, *338*, 1202–1205. [[CrossRef](#)]
16. Igarashi, J.; Shouno, O.; Fukai, T.; Tsujino, H. Real-time simulation of a spiking neural network model of the basal ganglia circuitry using general purpose computing on graphics processing units. *Neural Netw.* **2011**, *24*, 950–960. [[CrossRef](#)] [[PubMed](#)]
17. Simons, T.; Lee, D.-J. A Review of Binarized Neural Networks. *Electronics* **2019**, *8*, 661. [[CrossRef](#)]
18. Benjamin, B.V.; Gao, P.; McQuinn, E.; Choudhary, S.; Chandrasekaran, A.R.; Bussat, J.M.; Alvarez-Icaza, R.; Arthur, J.V.; Merolla, P.A.; Boahen, K. Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations. *Proc. IEEE* **2014**, *102*, 699–716. [[CrossRef](#)]
19. Deng, L.; Li, G.; Han, S.; Shi, L.; Xie, Y. Model Compression and Hardware Acceleration for Neural Networks: A Comprehensive Survey. *Proc. IEEE* **2020**, *108*, 485–532. [[CrossRef](#)]
20. Zuhra, K.; Nimet, K.; Yasemin, A.; Recai, K. An extensive FPGA-based realization study about the Izhikevich neurons and their bio-inspired applications. *Nonlinear Dyn.* **2021**, *105*, 3529–3549.
21. Price, J.B.; Rusheen, A.E.; Barath, A.S.; Cabrera, J.M.R.; Shin, H.; Chang, S.Y.; Kimble, C.J.; Bennet, K.E.; Blaha, C.D.; Lee, K.H.; et al. Clinical applications of neurochemical and electrophysiological measurements for closed-loop neurostimulation. *Neurosurg. Focus* **2020**, *49*, E6. [[CrossRef](#)] [[PubMed](#)]
22. Durand, D.M. Electric field effects in hyperexcitable neural tissue: A review. *Radiat. Prot. Dosim.* **2003**, *106*, 325–331. [[CrossRef](#)]
23. Chiang, C.C.; Shivacharan, R.S.; Wei, X.L.; Gonzalez-Reyes, L.E.; Durand, D.M. Slow periodic activity in the longitudinal hippocampal slice can self-propagate non-synaptically by a mechanism consistent with ephaptic coupling. *J. Physiol.* **2019**, *597*, 249–269. [[CrossRef](#)]
24. Yang, K.H.; Franaszczuk, P.J.; Bergey, G.K. The role of excitatory and inhibitory synaptic connectivity in the pattern of bursting behavior in a pyramidal neuron model. In Proceedings of the 8th International Conference on Neural Information Processing (ICONIP 2001), Shanghai, China, 14–18 November 2001; pp. 603–605.
25. Hodgkin, A.L.; Huxley, A.F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* **1952**, *117*, 500–544. [[CrossRef](#)]

26. Wei, X.; Xu, J.; Gong, B.; Chang, S.; Lu, M.; Zhang, Z.; Yi, G.; Wang, J. Multi-Core ARM-Based Hardware-Accelerated Computation for Spiking Neural Networks. *IEEE Trans. Ind. Inform.* **2023**, *19*, 8007–8017. [[CrossRef](#)]
27. Tisan, A.; Chin, J. An End-User Platform for FPGA-Based Design and Rapid Prototyping of Feedforward Artificial Neural Networks With On-Chip Backpropagation Learning. *IEEE Trans. Ind. Inform.* **2016**, *12*, 1124–1133. [[CrossRef](#)]
28. Wu, Q.X.; McGinnity, T.M.; Maguire, L.P.; Glackin, B.; Belatreche, A. Learning under weight constraints in networks of temporal encoding spiking neurons. *Neurocomputing* **2006**, *69*, 1912–1922. [[CrossRef](#)]
29. Yang, K.H.; Franaszczuk, P.J.; Bergey, G.K. The influence of synaptic connectivity on the pattern of bursting behavior in model pyramidal cells. *Neurocomputing* **2002**, *44*, 233–242. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.